Applied Mathematical Sciences

Wolfgang Hackbusch

Iterative Solution of Large Sparse Systems of Equations

Second Edition

Springer

Applied Mathematical Sciences

Volume 95

Wolfgang Hackbusch

Iterative Solution of Large Sparse Systems of Equations

Second Edition

Springer

Wolfgang Hackbusch
Max Planck Institute for Mathematics
in the Sciences
Leipzig
Germany

ISSN 0066-5452 ISSN 2196-968X (electronic)
Applied Mathematical Sciences
ISBN 978-3-319-28481-1 ISBN 978-3-319-28483-5 (eBook)
DOI 10.1007/978-3-319-28483-5

Library of Congress Control Number: 2016940360

Mathematics Subject Classification (2010): 65F10, 65N22, 65N55

Printed on acid-free paper

This Springer imprint is published by Springer Nature
The registered company is Springer International Publishing AG Switzerland

Preface

The numerical treatment of partial differential equations splits into two different parts. The first part are the discretisation methods and their analysis. This led to the author's monograph *Theory and Numerical Treatment of Elliptic Differential Equations* also published by Springer. The second part is the treatment of the equations obtained by the discretisation process. The arising system of linear (or even nonlinear) equations is of large size, only bounded by the available storage of the computers. Nowadays, systems of several millions of equations and variables must be solved. Another characteristic of the arising systems is the sparsity of the system matrix; i.e., only $\mathcal{O}(n)$ entries of the $n \times n$ matrix are different from zero. The classical Gauss elimination requires up to $\mathcal{O}(n^3)$ operations. Because of the large size of n, algorithms of this complexity are hopeless. Even methods requiring a cost of $\mathcal{O}(n^2)$ take a too long run time. Instead, one needs solution algorithms of complexity $\mathcal{O}(n)$ or $\mathcal{O}(n \log^* n)$.

This book grew out of a series of lectures given by the author at the Christian Albrecht University of Kiel to students of mathematics. The first German edition was published in 1991 by Teubner, Stuttgart. The second German edition in 1993 mainly corresponds to the first English edition at Springer in 1994. Since that time new methods have developed. Therefore the present second edition differs significantly from the first one.

Although special attention is devoted to the modern effective algorithms (multigrid iterations, domain decomposition methods, and the hierarchical LU iteration), the theory of classical iterative methods should not be neglected. One reason is that these iterations indirectly re-appear in modern methods.

This volume requires basic mathematical knowledge in analysis and linear algebra. The necessary facts from linear algebra and matrix theory are summarised in the Appendices A–C of this book in order to provide as complete a presentation as possible and present a formulation and notation needed here. Similarly, the basics of finite element discretisation are summarised in Appendix E.

Part I covers the introduction and the classical linear iterations. Part II describes the semi-iterative methods including the popular conjugate gradient method. The subjects of these two parts should be understood as two orthogonal methods: a linear iteration is accelerated by a semi-iterative approach. Part III contains more recent linear iterations.

The new Chapter 5 in Part I is devoted to the algebra in the set of linear iterations. These operations are important for the generation of new iterations. Part III contains two new chapters. Chapter 13 describes the $\mathcal{H}$-LU iteration which is based on the technique of hierarchical matrices introduced in Appendix D. In many cases, this iteration is a very efficient and robust method of black-box type. Finally, in Chapter 14, tensor-based iterative methods are briefly mentioned.

The discussion of the various methods is illustrated by many numerical examples, mostly for the Poisson model problem. Since these calculations are taken from the first edition, the problem sizes are small compared with modern computers. However, these sizes are completely sufficient to demonstrate the asymptotic behaviour.

The author also wishes to express his gratitude to the publisher Springer for their friendly cooperation.

Leipzig and Molfsee, October 2015 *Wolfgang Hackbusch*

Contents

List of Symbols and Abbreviations

Symbols

$\mathbf{1}$	vector $(1, 1, ..., 1)^{\mathsf{T}}$
$A^{\mathsf{T}}, A^{\mathsf{H}}$	transposed and Hermitian transposed matrix; cf. §A.1
$A^{-\mathsf{T}}, A^{-\mathsf{H}}$	inverse of $A^{\mathsf{T}}, A^{\mathsf{H}}$; cf. §A.1
$W^{\perp}$	orthogonal complement of a W; cf. §A.5
$U \oplus V$	direct sum of subspaces; cf. §A.5.3
$M\vert_b$	restriction of the matrix to the block b; cf. Notation D.6
$M\vert^b$	extension of the matrix to the block b; cf. §D.3.4
$\varDelta$	Laplace operator; cf. (1.1a)
$\langle \cdot, \cdot \rangle$	(Euclidean) scalar product; cf. (1.1a–c)
$\langle \cdot, \cdot \rangle_A$	energy scalar product; cf. (C.5b)
$\Vert\cdot\Vert$, $\vert\vert\vert \cdot \vert\vert\vert$	norm (of vectors or matrices)
$\Vert\cdot\Vert_A$	energy norm; cf. (C.5a)
$\Vert\cdot\Vert_2$	Euclidean norm, cf. (B.2); spectral norm, cf. (B.21a)
$\Vert\cdot\Vert_\infty$	maximum norm, cf. (B.2); row sum norm, cf. (B.8)
$\Vert\cdot\Vert_{Y\leftarrow X}$	norm of a mapping (matrix) from X into Y; cf. (B.11)
$\vert\vert\vert \cdot \vert\vert\vert_T$	transformed vector or matrix norm; cf. (B.10a,b)
$\vert\cdot\vert$	absolute value, in §C.3 applied to matrices and vectors; cf. page 438
$<, \leq, >, \geq$	in connection with matrices, the order relation from §C.1.2; only in §§C.3–C.4 (and §7.3.5) it denotes the order relation of (C.9a,b)
$\dot{\cup}$	disjoint union
$\subset$	$A \subset B$: A is a subset of B, not necessarily a proper subset
$\subsetneqq$	$A \subsetneqq B$: A is a proper subset of B
$\otimes$	tensor product; cf. §14.1
$\odot$	Hadamard product; cf. Lemma 5.60c
$\oplus_r$	addition of hierarchical matrices with truncation to rank r; cf. p. 457
$\odot_r$	multiplication of $\mathcal{H}$-matrices with truncation to rank r; cf. §D.3.4
$\circ$	product $\varPhi \circ \varPsi$ of iterations or transformation of iterations; cf. §5
$\#S$	cardinality of a set S

Greek Letters

α, β, γ	indices of the index set; cf. §1.3
γ	in §11: number of secondary multi-grid steps for the coarse-grid equation; cf. (11.33d$_2$)
γ, Γ	lower and upper eigenvalue bounds of $W^{-1}A$; cf. (9.18a)
δ_{ij}	Kronecker symbol: $\delta_{ij} = 1$ for $i = j$, $\delta_{ij} = 0$ otherwise
Δ	Laplace operator; cf. (1.1a)
ζ	often contraction number; cf. §2.2.6, (11.30b), (11.48)
η	characteristic factor involved in the admissibility condition (D.10)
$\eta(\nu)$	zero sequence for smoothing property; cf. (11.58b)
$\eta_0(\nu)$	special function, defined in Lemma 11.23
ϑ, Θ	damping factor; cf. §3.2.1 and §5.2
$\kappa(A)$	spectral condition number (B.13)
λ, Λ	eigenvalue bounds of A; cf. Theorem 9.10, Theorem 3.30
$\lambda_{\max}(A)$	maximal eigenvalue of a matrix A if $\sigma(A) \subset \mathbb{R}$
$\lambda_{\min}(A)$	minimal eigenvalue of a matrix A if $\sigma(A) \subset \mathbb{R}$
ν, ν_1, ν_2	in §11: number of the smoothing steps; cf. (11.21) and (11.22a)
$\rho(A)$	spectral radius of a matrix A; cf. Definition A.17
$\rho_{m+k,m}$	convergence factors; cf. (2.23a,b)
$\sigma(A)$	spectrum of the matrix A; cf. §A.3
τ, σ	symbols representing clusters; cf. §D.2.1
$\Phi(x, b, A)$	function describing an iteration; cf. (2.3)
$\Upsilon[\Phi]$	semi-iterative method with Φ as basic iteration
$\Upsilon_{a,b}^{\mathrm{Cheb}}$	Chebyshev method; cf. Notation 8.29
$\Upsilon_{\mathrm{CG}}, \Upsilon_{\mathrm{CR}}, \Upsilon_{\mathrm{OD}}, \Upsilon_{\mathrm{GMRES}}$	conjugate gradient methods and variants; cf. §10.2
Υ_{grad}	gradient method; cf. §9.2.1 and §9.2.4
ω	relaxation parameter; cf. (1.22) and §3.2.4
Ω	underlying domain of a boundary value problem; cf. (1.1a), §E.1
Ω_h	grid; cf. (1.3)

Latin Letters

$a(\cdot,\cdot)$	bilinear or sesquilinear form; cf. Definition (E.1)
a, b	bounds for $\sigma(M)$; cf. (8.26a)
A, A_ℓ	matrix of the linear system; cf. (1.5), (11.6a)
$A^{\kappa\lambda}, A^{ij}$	block of A; cf. (A.8b,c)
$A_{\alpha\beta}, a_{\alpha\beta}, A_{ij}, a_{ij}$	entries of the matrix A
b, b_ℓ	right-hand side of the linear system; cf. (1.5), (11.6a)
blockdiag$\{\ldots\}$	block-diagonal matrix; cf. page 408
blockdiag$_{\mathrm{B}}\{A\}$	block-diagonal part of A with respect to the block structure B; cf. (4.2$'$)
blocktridiag$\{\ldots\}$	block-tridiagonal matrix; cf. (A.9)

$\mathbb{C}$	complex numbers
$\mathrm{cond}, \mathrm{cond}_2$	condition of a matrix; cf. (B.12)
d^m	defect $Ax^m - b$; cf. (2.17)
$D, D', \ldots$	(block-) diagonal matrix
$\mathfrak{D}(\Phi)$	domain of the iteration Φ; cf. Definition 2.2a
$\deg_X(v)$	degree of a vector v; cf. Definition 8.10
$\mathrm{degree}(\cdot)$	degree of a polynomial
$\mathrm{depth}(T)$	depth of the tree T; cf. (D.7)
$\det$	determinant
$\mathrm{diag}\{\ldots\}$	diagonal matrix or diagonal part; cf. (A.1)
$\mathrm{diam}(\tau)$	diameter of a cluster; cf. (D.9a)
$\mathrm{dist}(\tau, \sigma)$	distance between clusters; cf. (D.9b)
e^m	error $x^m - x$ of the m-th iterate; cf. (2.15)
$\mathbf{e}_\alpha$	unit vector
E	strictly lower triangular matrix; cf. (1.16)
$\mathrm{Eff}(\Phi)$	effective amount of work; cf. (2.31a)
F	strictly upper triangular matrix; cf. (1.16)
$\mathcal{F}$	matrices in full format; cf. Definition D.2a
$G(A)$	graph of a matrix A; cf. Definition C.12
h, h_ℓ	grid size; cf. (1.2)
$\mathcal{H}^2$	see §D.2.7
$\mathcal{H}_p$	model format; cf. §D.1.3
i, j, k	indices of the ordered index set $I = \{1, \ldots, n\}$
I	identity matrix
I	index set (not necessarily ordered)
I_κ	subset of block indices; cf. (A.7)
$\mathit{Init}(\Phi, A)$	cost for initialising the iteration Φ applied to the system $Ax = b$
$\mathit{It}(\Phi)$	cf. (2.30a)
$\mathbb{K}$	the field $\mathbb{R}$ or $\mathbb{C}$
$\mathbb{K}^I$	space of the vectors corresponding to the index set I
$\mathbb{K}^{I \times I}$	space of the matrices corresponding to the index set I
$\mathcal{K}$	integral operator; cf. §D.2.9
$\mathcal{K}_m(X, v)$	Krylov space; cf. Definition 8.7
$\ker$	kernel of a mapping or matrix
ℓ	level number in the discretisation hierarchy; cf. (3.15a)
$L, L', \hat{L}$	lower (block-)triangular matrix
$\mathcal{L}$	set of consistent linear iterations; cf. (2.11)
$\mathcal{L}_{\mathrm{pos}}$	set of positive definite iterations; cf. Definition 5.8
$\mathcal{L}_{\mathrm{semi}}$	set of positive semidefinite iterations; cf. Definition 5.11
$\mathcal{L}_{\mathrm{sym}}$	set of symmetric iterations; cf. Definition 5.3
$\mathcal{L}_{>}$	set of directly positive definite iterations; cf. Definition 5.14
$\mathcal{L}(T)$	set of leaves of the tree T; cf. §D.2.1
$\mathrm{level}(\tau)$	level-number of a cluster; cf. §D.2.1
$\log$	natural logarithm
$\log_2$	dual logarithm, logarithm with respect to the basis 2

$\log^*(\cdot)$	some power of $\log(\cdot)$; cf. Footnote 9 on page 15
m	iteration number; cf. e^m, x^m
M, M^{xyz}	iteration matrix (of the iteration 'xyz'); cf. §2.2.1
$M[A]$	iteration matrix for the system $Ax = b$; cf. Definition 2.9
n, n_ℓ	dimension of the linear system; cf. §2.3, (11.6b)
$n_{\min}$	minimal size of clusters; cf. §D.2.1
N	number of the grid points per row or column; cf. (1.2)
N, N^{xyz}	matrix of the 2nd normal form (of the iteration 'xyz'); cf. (2.10)
$N[A]$	matrix N for the system $Ax = b$; cf. Definition 2.9
$\mathbb{N}$	natural numbers $\{1, 2, 3, \ldots\}$
$\mathbb{N}_0$	$\mathbb{N} \cup \{0\} = \{0, 1, 2, \ldots\}$
N_{xyz}	number of arithmetic operations required for 'xyz'; cf. pages 455ff
$\mathcal{O}(\cdot)$	Landau symbol: $f(\alpha) = \mathcal{O}(g(\alpha))$ if $\lvert f(\alpha)\rvert \le C \lvert g(\alpha)\rvert$ for the underlying limit process $\alpha \to 0$ or $\alpha \to \infty$. The notation $f(\eta) = 1 - \mathcal{O}(\eta^\tau)$ is more special and means that $f(\eta) \le 1 - C\eta^\tau$ with fixed $C > 0$ for $\eta \to 0$.
p	prolongation; cf. §11.1.3, (12.7)
P	partition of a hierarchical matrix; cf. §D.2.3
P^+, P^-	subsets of the partition P; cf. §D.2.6
$\mathcal{P}_m$	space of polynomials; cf. Definition 8.2
Q	often unitary matrix
$Q_{\min}(\cdot)$	bounding box; cf. §D.2.1.2
r	restriction; cf. §11.1.4, (12.14a)
r	representation rank of matrices in $\mathcal{R}_r$; cf. (D.2)
$r(A)$	numerical radius of the matrix A; cf. §B.3.4
$\mathbb{R}$	real numbers
$\mathrm{range}(\cdot)$	range (image space) of a mapping
$\mathcal{R}_r$	rank-r matrices or tensors; cf. Definition D.2b and page 390
$\mathrm{root}(T)$	root of the tree T; cf. §D.2.1
S_ℓ	iteration matrix of the smoother $\mathcal{S}_\ell$; cf. Lemma 11.11
$\mathcal{S}_\ell$	smoothing iteration; cf. §11.1.1 and §11.2.1
$\mathrm{span}\{\ldots\}$	linear space spanned by $\{\ldots\}$
$\mathrm{supp}(\cdot)$	support of a function; Footnote 6 on page 463
T_ℓ, T_r	left- and right-sided transformation; cf. (5.32), (5.39)
$T(I)$	cluster tree corresponding to the index set I; cf. §D.2.1
$T^{(\ell)}(I)$	subset of $T(I)$; cf. (D.7)
$T(I \times J)$	block cluster tree corresponding to the index set I; cf. §D.2.1
$\mathcal{T}_r$, $\mathcal{T}_r^{\mathcal{R}}$, $\mathcal{T}_r^{\mathcal{H}}$, $\mathcal{T}_{r,\mathrm{pairw}}^{\mathcal{R}}$	truncation operator; cf. §D.3.2
tridiag{...}	tridiagonal matrix; cf. (A.2)
u_{ij}	components of the grid function u; cf. (1.6b)
U, U', $\hat{U}$	upper (block-)triangular matrix
W, W_Φ	matrix of the third normal form (of the iteration Φ); cf. (2.12)
$W[A]$	matrix W for the system $Ax = b$; cf. Definition 2.9
Work(Φ, A)	amount of work of the iteration Φ applied to $Ax = b$; cf. (2.29)
x	vector; often solution of the equation $Ax = b$

x^*	solution of the equation $Ax = b$ if the symbol x is used as a variable
x_ℓ, x_ℓ^*	vectors x, x^* at the level ℓ; cf. (11.6a)
x^0	starting value of the iteration
x^m	m-th iterate
x^α, x^i	block of x corresponding to the index α or i; cf. (A.8a)
x_α, x_i	components of a vector x
x, y	spatial variables $(x, y) \in \Omega$; cf. (1.1a)
X_τ	support of the cluster τ; cf. §D.2.1.2 and (D.8)
$\mathbb{Z}$	set of integers

Abbreviations and Algorithms

ALS	alternating least squares method cf. §14.4
AMG	algebraic multigrid method
AMLI	algebraic multilevel iteration; cf. page 335
AOR	accelerated overrelaxation
ART	algebraic reconstruction technique
BCG, BiCG	biconjugate gradient method; cf. §10.5.3
BEM	boundary element method
Bi-CGSTAB	biconjugate gradient stabalised method; cf. §10.5.3
BPX	additive multigrid iteration; cf. §12.9.6
CG	method of congujate gradients; cf. §10.2
CGS	conjugate gradient squared method; cf. §10.5.3
CR	method of congujate residuals; cf. §10.3
DDM	domain decomposition method
FEM	finite element method
FFT	fast Fourier transform
FOM	full orthogonalisation method; cf. §10.5.2
GMRES	generalised minimal residual method; cf. §10.5.1
$\mathcal{H}$-matrix	hierarchical matrix
$\mathcal{H}$-LU	hierarchical LU decomposition
HOSVD	higher order singular value decomposition; cf. §14.2.3
MAOR	modified accelerated overrelaxation
MINRES	minimal residual method
MSOR	modified successive overrelaxation
OD	method of orthogonal directions; cf. §10.4
ORTHODIR, ORTHOMIN, ORTHORES	cf. §10.5.4
SAOR	symmetric accelerated overrelaxation
SIRT	simultaneous iterative reconstruction technique; cf. page 94
SOR	successive overrelaxation; cf. §3.2.4
SVD	singular value decomposition; cf. §A.6.4
SYMMLQ	symmetric LQ method; cf. page 257
USSOR	unsymmetric successive overrelaxation

Part I
Linear Iterations

The core of iterative methods for linear systems are *linear iterations*. Different from direct methods, an infinite sequence of iterates is produced. Since, in practice, only a finite number of iteration steps is performed, the unavoidable iteration error depends crucially on the speed of convergence.

Chapter 1 starts with historical remarks. It introduces the Poisson model problem which will be a test example for the iterative methods described later on. Vector and matrix notations are provided in §1.3. In the case of discretisations of partial differential equations, it is important to consider the family of systems obtained for different discretisation parameters (§1.4). A crucial question is whether the convergence speed deteriorates with increasing matrix size. To get a first idea of an iterative method, the Gauss–Seidel and SOR methods are presented in §1.6 with numerical results for the model problem. These examples involve sparse matrices (§1.7). Except for Chapter D, we shall always assume that the underlying matrices are sparse. This assumption ensures that the cost of one iteration step is proportional to the matrix size; however, sparsity is not needed for convergence analysis.

Chapter 2 introduces general iterative methods. The concepts of consistency and convergence are described in §2.1. The class of linear iterations is specified in §2.2. For its description three normal forms are introduced. A first important result is the convergence theorem in §2.2.5. The quality of a linear iteration depends on both cost and convergence speed. The resulting efficacy is discussed in §2.3. Section 2.4 demonstrates how to test iterative methods numerically.

The convergence of a linear iteration depends on the properties of the underlying matrix. Chapter 3 investigates classical iterations (Richardson, Jacobi, Gauss–Seidel, SOR) applied to positive definite matrices. The corresponding analysis of general linear iterations is presented in Chapter 6.

Chapter 4 considers classical iterations assuming other structural matrix properties. In particular, §4.6 contains Young's theorem on SOR for consistently ordered matrices. It describes the improvement of the convergence order in explicit form.

The set of linear iterations forms an algebra containing various operations as described in Chapter 5. Section 5.1 introduces the definition of an adjoint iteration. This enables the construction of symmetric or even positive definite iterations. Damping of linear iterations is discussed in §5.2. Addition of linear iterations is the subject of §5.3, while the product of linear iterations is investigated in §5.4. Another combination of iterative methods is the secondary iteration (§5.5). The left, right, or two-sided transformations are studied in §5.6. Kaczmarz' iteration (§5.6.3) and Cimmoni's iteration (§5.6.4) can be obtained by suitable transformations.

Chapter 6 collects the convergence results for positive definite iterations (including possible perturbations of the positive definite matrix). In particular, the symmetric Gauss–Seidel method and symmetric SOR are studied (§6.3).

Chapter 7 is concerned with the generation of linear iterations. A classical technique is additive splitting of the underlying matrix (§7.2). Incomplete LU decomposition (ILU, §7.3) is another possibility to generate an iteration by matrix data only. Preconditioning in §7.4 is a particular case of a transformation aiming at improving the convergence.

Modern linear iterations will be treated in Part III.

Chapter 1
Introduction

I recommend this [iterative] method to you for imitation. You will hardly ever again eliminate directly, at least not when you have more than 2 unknowns. The indirect procedure can be done while half asleep, or while thinking about other things.

(C.F. Gauss in a letter to Gerling [148], Dec. 1823).

Abstract After some historical comments in Section 1.1, we introduce a model problem (Section 1.2) serving as a first test example of the various iterative methods. Deliberately, a simply structured problem is chosen since this allows us to determine all required quantities explicitly. The role of the ordering of the unknowns is explained. Often no ordering is needed. Section 1.3 introduces notation for vectors and matrices. Furthermore, the description of difference schemes by stencils is explained. Besides the behaviour of an iterative method for a single system, its behaviour with respect to a whole family of systems is often more interesting (Section 1.4). In Section 1.5, the cost of the direct solution by the Gauss elimination is determined. This cost can be compared with the cost of the iterative methods introduced later. In Section 1.6, the Gauss–Seidel and SOR iteration are presented as first examples of linear iterations. Finally, in Section 1.7, sparsity of the underlying matrix discussed.

1.1 Historical Remarks Concerning Iterative Methods

Iterative methods are almost 200 years old. The first iterative method for systems of linear equations is due to Carl Friedrich Gauß (simplified spelling: Gauss). His method of least squares led him to a system of equations that was too large for the use of direct Gauss elimination.[1] Today the iterative method described in Gauss [147][2] would be called the blockwise Gauss–Seidel method. The value that Gauss attributed to his iterative method can be seen in the excerpt from his letter [148][3] at the top of the page.

[1] The Gauss elimination is known since ancient times; the Chinese text *Jiu Zhang Suanshu: Nine Chapters on the Mathematical Art* is written about 200 BC.

[2] A translation of the neo-Latin title is 'Supplement to the theory on the combination of observations subject to minimal errors'.

[3] See also the English translation by Forsythe [137].

Carl Gustav Jacobi [227][4] described a very similar method in 1845. In 1874 Phillip Ludwig Seidel, a student of Jacobi, wrote about 'a method, to solve the equations arising from the least squares method as well as general linear equations by successive approximation' [337].

Since the time that electronic computers became available for solving systems of equations, the number of equations has increased by many orders of magnitude and the methods mentioned above have proved to be too slow. After more than 100 years of stagnation in this field, Southwell [346, 347, 348, 349] experimented with variants of the Gauss–Seidel method[5] and, in 1950, David M. Young, Jr. [411] succeeded in a breakthrough. His modification of the Gauss–Seidel method leads to an important acceleration of the convergence. This so-called SOR iteration will be described in §1.6 as an example of an iterative method. Since then, numerous other methods have been developed. The modern ones will be described in Part III.

Concerning a historical view to the development of iterative techniques, we recommend, e.g., the articles by Stiefel [353] (1952), Forsythe [138] (1953), Axelsson [10, §7.1] (1976), and Young [413] (1989).

1.2 Model Problem: Poisson Equation

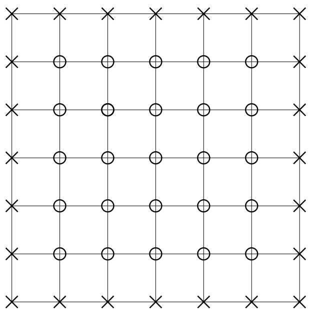
Fig. 1.1 Grid Ω_h with inner grid points (o) and boundary points ($\times$).

During the time of Gauss, Jacobi and Seidel, the equations of the least squares method have led to a larger number of equations (e.g., obtained from geodesic measurements). Today, in particular the discretisations of partial differential equations give rise to systems of a large number of equations.

Since the discretisation error is smaller the larger the dimension of the system is, one is interested in systems of millions of unknowns[6]. In the following we shall often refer to a model problem representing the simplest nontrivial example of a boundary value problem. It is the Poisson equation with Dirichlet boundary values:

$$-\Delta u(x,y) = f(x,y) \qquad \text{for } (x,y) \in \Omega, \tag{1.1a}$$
$$u(x,y) = \varphi(x,y) \qquad \text{on } \Gamma = \partial\Omega. \tag{1.1b}$$

[4] The English translation of the title is 'On a new solution method of linear equations arising from the least squares method'.

[5] Southwell used the term *relaxation*, since he considered the system of equations as a mechanical arrangement. The solution of the system characterises the equilibrium. Otherwise, forces act between nodes. One partial step of the Gauss–Seidel method leads to the local equilibrium in one node, i.e., this node is *relaxed*.

[6] The concrete size is time dependent, since it increases with the available computer capacity.

Here $\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$ abbreviates the two-dimensional[7] Laplace operator. As the underlying domain Ω, we choose the unit square

$$\Omega = (0,1) \times (0,1). \tag{1.1c}$$

In (1.1a,b), the *source term* f and the *boundary values* φ are given, while the function u is unknown.

To discretise the differential equation (1.1a–c), the domain Ω is covered with a grid of step size h (cf. Figure 1.1). Each grid point (x, y) has the representation $x = ih,\ y = jh\ (0 < i, j < N)$, where

$$h = 1/N. \tag{1.2}$$

More precisely, the grid is the set of *inner* grid points:

$$\Omega_h := \{(x, y) = (ih, jh) : 1 \le i, j \le N - 1\}. \tag{1.3}$$

We abbreviate the desired values $u(x, y) = u(ih, jh)$ with u_{ij}. An approximation of the differential equation (1.1a) is given by the *five-point formula*

$$h^{-2} \left[4u_{ij} - u_{i-1,j} - u_{i+1,j} - u_{i,j-1} - u_{i,j+1} \right] = f_{ij} \tag{1.4a}$$

with $f_{ij} := f(ih, jh)$ for $1 \le i, j \le N - 1$. The left-hand side in (1.4a) coincides with $-\Delta u(ih, jh)$ up to a consistency error $\mathcal{O}(h^2)$ when a sufficiently smooth solution u of (1.1a,b) is inserted (cf. Hackbusch [193, §4.5]). For grid values on the boundary, i.e., for $i = 0$, $i = N$, $j = 0$, or $j = N$, the values u_{ij} are known from the boundary data (1.1b):

$$u_{ij} := \varphi(ih, jh) \qquad \text{for } i = 0,\ i = N,\ j = 0, \text{ or } j = N. \tag{1.4b}$$

The number of the unknowns u_{ij} is $n := (N - 1)^2$ and corresponds of the number of the inner grid points. In order to form the system of equations, we have to eliminate the boundary values (1.4b), which possibly may appear in (1.4a). For instance, if $N \ge 3$, the equation corresponding to the index $(i, j) = (1, 1)$ reads as

$$h^{-2}[4u_{11} - u_{12} - u_{21}] = g_{11} \quad \text{with} \quad g_{11} := f(h, h) + h^{-2}[\varphi(0, h) + \varphi(h, 0)].$$

To write the equations in the common matrix formulation

$$Ax = b \tag{1.5}$$

with an $n \times n$ matrix A and n-dimensional vectors x and b with $n = (N - 1)^2$, one is forced to represent the doubly indexed unknowns u_{ij} by a singly indexed vector x. This implies that the (inner) grid points must be enumerated in some way.

[7] The one-dimensional Laplace equation $-u'' = f$ leads to a too simple system which is not suited as a test example. The two-dimensional problem has already the typical properties. The three-dimensional counterpart would not be better.

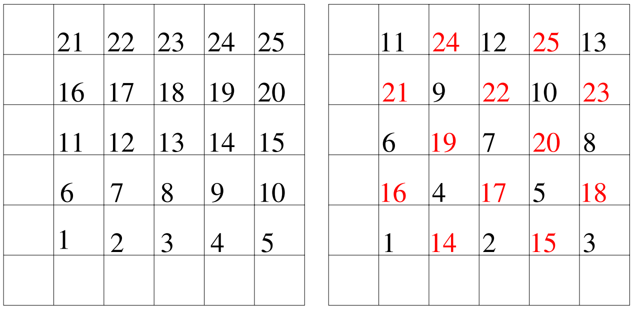

Fig. 1.2 *Left:* lexicographical ordering of the grid points. *Right:* chequer-board ordering.

Figure 1.2 (left) shows the lexicographical ordering. The exact definition of the matrix A and of the right-hand side b can be seen from the following definition of the matrix A and of the vector b by the lexicographical ordering for the Poisson model problem with step size $h = 1/N$:

```
$A := 0;$   {all entries of $A$ are initialised by zero}                         (1.6a)
$k := 0;$   {$1 \le k \le n$ is the index with respect to the lexicographical ordering}
for $j := 1$ to $N-1$ do for $i := 1$ to $N-1$ do
begin $k := k+1;\ a_{kk} := 4 \cdot h^2;\ b_k := f(ih, jh);$
    if $i > 1$     then $a_{k-1,k} := -h^2$      else $b_k := b_k + h^2 \cdot \varphi(0, jh);$
    if $i < N-1$   then $a_{k+1,k} := -h^2$      else $b_k := b_k + h^2 \cdot \varphi(1, jh);$
    if $j > 1$     then $a_{k,k-(N-1)} := -h^2$  else $b_k := b_k + h^2 \cdot \varphi(ih, 0);$
    if $j < N-1$   then $a_{k,k+(N-1)} := -h^2$  else $b_k := b_k + h^2 \cdot \varphi(ih, 1)$
end;
```

Vice versa, the solution x of $Ax = b$ has to be interpreted as

$$\begin{array}{c} x_k = u_{ij} = u(ih, jh) \text{ for} \\ k = i + (j-1)(N-1) \end{array} \qquad (1 \le i, j \le N-1). \tag{1.6b}$$

When x is interpreted as a grid function, we use the notation u_{ij} or $u(x, y)$ with $x = ih,\ y = jh$.

Remark 1.1. The reformulation of the two-dimensionally ordered unknowns into a one-dimensionally ordered vector is rather unnatural. The reason should not be sought in the two-dimensional nature of the problem, but rather in the questionable idea of enumerating the vector components by indices 1 to n. We shall see that the matrix A will never be required in the full presentation (1.6a).

If, nevertheless, one wants to represent the matrix A as $(a_{ij})_{1 \le i,j < N}$, A should be written as a block matrix. The vector x decomposes naturally into $N-1$ blocks

$$x^j := \begin{bmatrix} x_{k+1} \\ \vdots \\ x_{k+N-1} \end{bmatrix} = \begin{bmatrix} u_{1,j} \\ \vdots \\ u_{N-1,j} \end{bmatrix} \qquad \begin{array}{l} \text{with } k := (j-1)(N-1) \\ \text{for } j = 1, \ldots, N-1, \end{array} \tag{1.7}$$

corresponding to the j-th row in the grid Ω_h. Accordingly, A takes the form of a block-tridiagonal matrix built from $(N-1) \times (N-1)$ blocks T, which again are tridiagonal $(N-1) \times (N-1)$ matrices:

$$A = h^{-2} \begin{bmatrix} T & -I & & & \\ -I & T & -I & & \\ & \ddots & \ddots & \ddots & \\ & & -I & T & -I \\ & & & -I & T \end{bmatrix}, \qquad T = \begin{bmatrix} 4 & -1 & & & \\ -1 & 4 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 4 & -1 \\ & & & -1 & 4 \end{bmatrix}. \tag{1.8}$$

I is the $(N-1)\times(N-1)$ identity matrix. Unmarked matrix entries or blocks are zeros or zero blocks, respectively. The representation (1.8) proves the next remark.

Remark 1.2. For the lexicographical ordering of the unknowns, the matrix A has a block-tridiagonal structure.

The lexicographical ordering is by no means the only ordering one can think of. Another frequently used approach is the *chequer-board ordering* (cf. Fig. 1.2, right).

In that case, the components u_{ij} with an even sum $i + j$ ('black squares') are enumerated first and thereafter those with an odd sum $i + j$ ('red squares') are numbered lexicographically. In the course of the next chapters further orderings will be mentioned. A broad collection of orderings of practical interest is given by Duff–Meurant [117].

Exercise 1.3. In the case of the chequer-board ordering, A decomposes into two blocks corresponding to the 'red' and 'black' indices. Prove that A has the block structure (1.9) with a rectangular submatrix B and identity matrices I_r, I_b whose block sizes are given by the numbers of the red and black grid points:

$$A = \begin{bmatrix} D_r & B \\ B^\mathsf{T} & D_b \end{bmatrix}, \qquad D_r = 4h^{-2} I_r, \quad D_b = 4h^{-2} I_b\,. \tag{1.9}$$

1.3 Notation

1.3.1 Index Sets, Vectors, and Matrices

According to Remark 1.1, the indices of the vectors are considered as unordered (unless we refer explicitly to a particular ordering). The (always finite) index set is denoted by I. The elements of I are often denoted by Greek letters, e.g., $\alpha \in I$.

In the case of the model problem, the indices $\alpha \in I$ are either the pairs $\alpha = (i,j)$ of the integers $1 \leq i,j \leq N-1$ or the grid points $(x,y) = (ih,jh)$. We denote the *cardinality* of I, i.e., the number elements of I, by $\#I$.

In general, we use the field $\mathbb{C}$ of complex numbers. This includes the standard case of the real field $\mathbb{R}$. For real matrices, the Hermitian transposed matrix A^{H} may be replaced by A^{T}. The neutral notation $\mathbb{K}$ stands for $\mathbb{R}$ or $\mathbb{C}$:

$$\mathbb{K} \in \{\mathbb{R}, \mathbb{C}\}. \tag{1.10}$$

A vector $b \in \mathbb{K}^I$ is a mapping $b : I \to \mathbb{K}$ into the field $\mathbb{K}$. The value of b at $\alpha \in I$ is denoted as vector component b_α. In programs, the notation $b[\alpha]$ is often used. If the index is a pair, e.g., $\alpha = (i,j)$, we write $b_{i,j} = b[i,j]$. A vector, composed of its components b_α, is written in the form

$$b = (b_\alpha)_{\alpha \in I} .$$

If the index set is ordered, we identify the indices with $1, 2, \ldots, n := \#I$. While the indices $\alpha, \beta, \gamma, \ldots$ are used for nonordered indices, we use Latin letters $i, j, k, \ldots$ in the ordered case.

In general, subscripts indicate the components of a vector. Sometimes, a subscript enumerates vectors; e.g., the first column vector of a matrix A may be written as $\mathbf{a}_1$. In order to avoid confusion with vector components, indexed vectors will be written in boldface as in the previous example. If not defined differently, $\mathbf{e}_\alpha$ abbreviates the α-unit vector with the components $(\mathbf{e}_\alpha)_\beta = \delta_{\alpha\beta}$. Here,

$$\delta_{\alpha\beta} = \left\{ \begin{array}{l} 1 \text{ for } \alpha = \beta \\ 0 \text{ for } \alpha \neq \beta \end{array} \right\} \qquad (\alpha, \beta \in I) \tag{1.11}$$

is the Kronecker symbol.

Square matrices are mappings of the set $I \times I$ of index pairs into $\mathbb{K}$. The set of these matrices is denoted by $\mathbb{K}^{I \times I}$. Matrices are always symbolised by upper-case letters. The matrix entry of A corresponding to the index pair $(\alpha, \beta) \in I \times I$ is written as $a_{\alpha\beta}$ or $a_{\alpha,\beta}$, and occasionally as $A_{\alpha\beta}$. Alternatively, the notation $A[\alpha,\beta] = a[\alpha,\beta]$ is used. In particular, $(A+B)_{\alpha\beta}$, $(A^{-1})_{\alpha\beta}$, etc. is written for the components of matrix expressions. The matrix composed of the entries $a_{\alpha\beta}$ is denoted by

$$A = (a_{\alpha\beta})_{\alpha,\beta \in I}.$$

The symbol

$$I = (\delta_{\alpha\beta})_{\alpha,\beta \in I}$$

abbreviates the *identity matrix*, since it cannot be confused with the index set I.

In the case of rectangular (sub)matrices, the indices α and β belong to different sets I and J: $A = (a_{\alpha\beta})_{\alpha \in I, \beta \in J}$ is an $I \times J$ matrix. The set of these matrices is denoted by $\mathbb{K}^{I \times J}$.

For an ordered index set I, e.g., $I = \{1, \ldots, n\}$, we use the standard index notation a_{ij} or A_{ij}.

1.3.2 Star Notation

In §1.2 the index set $I = \Omega_h$ is used. In the following, Ω_h can be more general than in (1.3). It may be an arbitrary subset of the two-dimensional infinite grid $\{(x,y) = (ih,jh) : i,j \in \mathbb{Z}\}$. The vector $x \in \mathbb{K}^I$ can be interpreted as a *grid function*, i.e., of a mapping defined at the grid points. Since the letter x represents the vector as well as the first component in the point $(x,y) \in \Omega_h$, we write u instead of $x \in \mathbb{K}^I$ in accordance with the equations (1.1a,b):

$$x_\alpha = u(x,y) \qquad \text{for } \alpha = (x,y) \in I = \Omega_h. \tag{1.12}$$

If it seems to be more favourable, the argument $(x,y) = (ih,jh)$ is replaced with the indices 'ij':

$$u(ih,jh) = u_{ij} \qquad \text{for } (ih,jh) \in \Omega_h\,.$$

The first index component x or i corresponds to the grid row (oriented from left to right), the second component y or j to the grid column (from the bottom to the top).

Mappings (matrices) defined in $\mathbb{K}^I$ with $I = \Omega_h$ can conveniently be described by using the *star* or *stencil notation*. The *nine-point formula*

$$\begin{bmatrix} a_{-1,1} & a_{0,1} & a_{1,1} \\ a_{-1,0} & a_{0,0} & a_{1,0} \\ a_{-1,-1} & a_{0,-1} & a_{1,-1} \end{bmatrix} \tag{1.13a}$$

represents a matrix containing the nine coefficients a_{pq} $(-1 \le p,q \le 1)$ of (1.13a) in each row. The component of Ax associated with the index $(ih,jh) \in \Omega_h$ is

$$\sum_{p,q=-1}^{1} a_{pq}\, u_{i+p,j+q} \quad \text{or} \quad \sum_{p,q=-1}^{1} a_{pq}^{ij}\, u_{i+p,j+q}\,, \tag{1.13b}$$

where $u = x$ according to (1.12). In the left part of (1.13b), the matrix entries are independent of the grid point (as, e.g., for the Poisson model problem), whereas, in the right part, they depend on $(ih,jh) \in \Omega_h$. $a_{00} = a_{00}^{ij}$ is the diagonal element $A_{(ij),(ij)}$ corresponding to the index 'ij'. For example, the element $a_{1,0}$ in (1.13a) at the right position in the middle row is the matrix entry $A_{(i,j),(i+1,j)}$ by which the right neighbour $u_{i+1,j}$—corresponding to the grid point $((i+1)h, jh) \in \Omega_h$—has to be multiplied in (1.13b).

Although $(ih,jh) \in \Omega_h$, the index $(i+p, j+q)$ appearing in (1.13b)—more precisely the grid point $((i+p)h, (j+q)h)$—may not belong to Ω_h. In this case, the term $a_{pq}^{ij} u_{i+p,j+q}$ in (1.13b) has to be ignored. The same effect is obtained by the formal definition $u_{i+p,j+q} := 0$.

The *five-point formula* of the Poisson model problem is

$$h^{-2}\begin{bmatrix} & -1 & \\ -1 & 4 & -1 \\ & -1 & \end{bmatrix}. \tag{1.14}$$

Unmarked entries a_{pq} (as at the positions $p,q = \pm 1$ in (1.14)) are defined by zero.

1.4 A Single System Versus a Family of Systems

Usually, a discretisation matrix A is embedded into a family $\{A_h\}_{h\in H}$. Here H is an infinite set with accumulation point $0 \in \overline{H}$. For instance, the Poisson model problem is defined for all $N \in \mathbb{N}\backslash\{1\}$ and the corresponding step sizes $h := \frac{1}{N} \to 0$. Statements about the convergence speed may be of the form $1 - \mathcal{O}(h^\kappa)$ (i.e., $\leq 1 - Ch^\kappa$ for some fixed C). Such expressions only make sense if there is a limit process $h \to 0$.

Given a family $\{A_\eta\}_{\eta\in F}$ of matrices, one is interested in the behaviour of the convergence rates (or of the computational cost for obtaining a certain accuracy) with respect to a limit process $\eta \to 0$ (or $\eta \to \infty$). If the iteration method leads to convergence estimates which are uniform with respect to η, we say that the iteration method is *robust* with respect to $\eta \in F$. A standard parameter is the discretisation size $h \to 0$, but it is not the only one. For instance, increasing anisotropy can be described by $A_\eta := B + \eta C$ for $\eta \to 0$, where B and C are discretisations of $\partial^2/\partial x^2$ and $\partial^2/\partial y^2$, respectively. Increasing convection is modelled by $A_\eta := \eta B + C$ $(\eta \to 0)$, where C is a discretisations of a differential operator of first order.

1.5 Amount of Work for the Direct Solution of a Linear System

Methods are called *direct* if they terminate after finitely many operations with an exact solution (up to floating-point errors). The best known direct method is the Gauss elimination. In the case of the model problem in §1.2, one may perform this method without pivoting (cf. §C.4.4).

Concerning the valuation of the amount of computational work, we do not distinguish between additions, subtractions, multiplications, or divisions. Each is counted as one (arithmetic) operation. Traditionally, arithmetic operations for indices, data transfer, and similar activities are not counted (cf. Björck [48, §1.1.4]).

Remark 1.4. In the general case, the Gauss elimination solving a system $Ax = b$ of n equations requires $2n^3/3 + \mathcal{O}(n^2)$ operations. The storage amounts to $n^2 + n$.

Proof. During the i-th elimination step, the i-th row contains $n - i$ nonzero elements, whose multiples have to be subtracted from $n-i-1$ matrix rows. Summation of these $2(n-i)^2 + \mathcal{O}(n)$ operations over $1 \leq i \leq n$ yields the statement. □

In the model case, $n = (N-1)^2 = h^{-2} + \mathcal{O}(h^{-1})$ implies the following.

Conclusion 1.5. *A naive application of the Gauss elimination to the model problem in §1.2 leads to* $2N^6/3 + \mathcal{O}(N^5) = 2h^{-6}/3 + \mathcal{O}(h^{-5})$ *operations and requires storage of* $N^4 + \mathcal{O}(N^3) = h^{-4} + \mathcal{O}(h^{-3})$.

Halving the grid size h, yields the 64-fold computational work. Assuming one second for the solution of grid size h, the same computation for the quartered grid size $h/4$ consumes more than one hour!

However, the amount of work is less if the system matrix $A \in \mathbb{R}^{n\times n}$ is a *band matrix*. Here we assume the ordered index set $I = \{1, \ldots, n\}$.

Definition 1.6. A is a band matrix of band width $w \in \mathbb{N}_0$ if $a_{ij} = 0$ holds for all $|i - j| > w$.

A band matrix has at maximum $2w$ nonvanishing off-diagonals besides the main diagonal. Concerning the properties of band matrices, we refer to Berg [44].

Remark 1.7. The matrix A arising from the model problem with lexicographical ordering according to (1.7) is a band matrix of band width $w = N - 1$.

The major part of the amount of work given in Remark 1.4 consists of unnecessary multiplications and additions by zeros. During the i-th elimination step the i-th row contains $w + 1$ nonzero elements. It is sufficient to eliminate the next w rows. This leads to $2w^2$ operations. In total, one obtains the next result.

Remark 1.8. The amount of work for the Gauss elimination without pivoting for solving a system with an $n \times n$ matrix of band width w amounts to

$$2nw^2 + \mathcal{O}(nw + w^3).$$

The storage requirement reduces to $2n(w + 1)$ when only the $2w + 1$ diagonals of A and the right-hand side b are stored.

Conclusion 1.9. *In the case of the model problem in §1.2, w is equal to $N - 1$. Therefore, the banded Gauss elimination requires $2N^4 + \mathcal{O}(N^3) = h^{-4} + \mathcal{O}(h^{-3})$ operations and storage of $2N^3 + \mathcal{O}(N^2)$.*

In the latter version, $2w + 1$ diagonals of A are used, although the matrix A in (1.8) has only five diagonals: the main diagonal, two side-diagonals at distance 1, and two further ones at distance $N - 1$. Unfortunately, one cannot exploit this property for the Gauss elimination.

Remark 1.10. The zeros in the second to $(N-2)$-th side-diagonals of the matrix A in (1.8) are completely filled during the elimination process by nonzeros (with the exception of the first block).

This occurrence is called *fill-in* and indicates a principal disadvantage of Gauss elimination when applied to sparse matrices. Here, we call an $n \times n$ matrix *sparse*, if the number of nonzero entries is by far smaller than n^2. Otherwise, the matrix is called a *fully populated* or *dense* matrix. Because of the equivalence of Gauss elimination to the triangular or LU decomposition (cf. Quarteroni–Sacco–Saleri [314, §3], Björck [48, §1.2]), the same difficulties holds for the LU decomposition.

Conclusion 1.11. *The decomposition $A = LU$ into a lower triangular matrix L and an upper triangular matrix U for the sparse matrix A in (1.8) yields factors L and U, which are full band matrices of width $w = N - 1$. The same holds for Cholesky decomposition.*

There are special direct methods solving the system described in §1.2 with an amount of work between $\mathcal{O}(n) = \mathcal{O}(N^2)$ and $\mathcal{O}(n \log n) = \mathcal{O}(N^2 \log N)$. Examples are the *Buneman algorithm* and the *method of total reduction*, both described in Meis–Marcowitz [281, 282] (see also Bank [25], Bjørstad [49], Buneman [87], Buzbee et al. [90], Duff–Erisman–Reid [116], Golub [154], Hockney [223], and Schröder–Trottenberg [333]).

1.6 Examples of Iterative Methods

For the iterative solution of a system, one starts with an arbitrary starting vector x^0 and computes a sequence of iterates x^m for $m = 1, 2, \ldots$:

$$x^0 \longmapsto x^1 \longmapsto x^2 \longmapsto \ldots \longmapsto x^m \longmapsto x^{m+1} \longmapsto \ldots$$

In the following, x^{m+1} is only dependent on x^m, so that the mapping $x^m \mapsto x^{m+1}$ determines the iteration method. The choice of the starting value x^0 is not part of the iteration method.

The already mentioned Gauss–Seidel iteration for solving the system $Ax = b$ reads as follows:

$$\textbf{for } i := 1 \textbf{ to } n \textbf{ do } x_i^{m+1} := \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{m+1} - \sum_{j=i+1}^{n} a_{ij} x_j^m \right) / a_{ii}. \tag{1.15}$$

Remark 1.12. (a) The Gauss–Seidel iteration (1.15) can be performed whenever all diagonal entries satisfy $a_{ii} \neq 0$.
(b) During the execution of the iteration, the variable x_i^m may be overwritten by the new value x_i^{m+1}.
(c) Different orderings (e.g., lexicographical or chequer-board ordering) yield different results.

Each matrix A can uniquely be decomposed into the sum

$$A = D - E - F, \qquad \left\{ \begin{array}{l} D \text{ diagonal matrix,} \\ E \text{ strictly lower triangular matrix,} \\ F \text{ strictly upper triangular matrix.} \end{array} \right\} \tag{1.16}$$

Here, E is called a *lower triangular matrix* if $E_{ij} = 0$ for $j > i$, and a *strictly lower triangular matrix*, if $E_{ij} = 0$ for $j \geq i$. The (strictly) upper triangular matrix is defined analogously. The system of equations $Ax = b$ is equivalent to

$$(D - E)x = b + Fx. \tag{1.17}$$

Replacing x by x^m on the right-hand side and by x^{m+1} on the left-hand side, we obtain the iterative description (1.18a) or (1.18b):

$$(D - E)x^{m+1} = b + Fx^m, \qquad \text{i.e.,} \tag{1.18a}$$

$$x^{m+1} = (D - E)^{-1}(b + Fx^m). \tag{1.18b}$$

Exercise 1.13. Prove that (1.18a,b) and (1.15) are equivalent, i.e., (1.18a) and (1.18b) are the vector representations of the Gauss–Seidel iteration, while (1.15) is the componentwise representation.

For a sparse matrix, one has to avoid defining A, b by (1.6a) and applying (1.15). For the model problem (1.4a,b), we should use the original data $f_{ij} = f[i,j]$ in (1.4a) and the boundary data $\varphi(ih, jh) = u_{ij} = u[i,j]$, which are stored at the boundary points of the array u. According to (1.6b), we use the variables u and f instead of x and b. The lexicographical Gauss–Seidel method for the model problem then takes the following form:

```
procedure GaussSeidel($u, f$);                                                    (1.19)
begin for $j := 1$ to $N - 1$ do for $i := 1$ to $N - 1$ do
    $u[i,j] := (h^2 \cdot f[i,j] + u[i-1,j] + u[i+1,j] + u[i,j-1] + u[i,j+1])/4$
end; {lexicographical ordering}
```

In the double loop of (1.19), the matrix A is explicitly represented by its nonzero entries. The indexing is based on the 'natural' double indices. The lexicographical ordering of the grid points is a consequence of the arrangement of the loops. For the chequer-board ordering, the loop in (1.19) can be changed as follows:

```
$w := 2$; {red squares} for $j := 1$ to $N - 1$ do                                (1.20)
begin $w := 3 - w$; for $i := w$ step 2 to $N - 1$ do
        $u[i,j] := (h^2 \cdot f[i,j] + u[i-1,j] + u[i+1,j] + u[i,j-1] + u[i,j+1])/4$
end;
$w := 1$; {black squares} for $j := 1$ to $N - 1$ do ... {same loop as in lines 2–4}
```

Since one may immediately store $h^2 \cdot f[i,j]$ instead of $f[i,j]$, the next remark follows.

Remark 1.14. In the case of the model problem, the Gauss–Seidel method (independently of the ordering) requires $5n$ operations ($4n$ additions and n divisions) per iteration.

Remark 1.15. In the case of $f_{ij} = -4$ and $\varphi(x,y) = x^2 + y^2$, the corresponding solution is $u_h(x,y) = x^2 + y^2$, i.e., $u_{ij} = (i^2 + j^2)h^2$. The simplest starting values are $u_{ij}^0 = 0$. In the following, the system (1.18a) with these data will be called the *Poisson model problem*. In the course of the next chapters, various iterative methods will be tested using this example.

Table 1.1 shows the error

$$\varepsilon_m := \max\left\{ \left|u_{ij}^m - (i^2+j^2)h^2\right| : 1 \le i,j \le N-1 \right\}$$

of the m-th iterate for $h = \frac{1}{32}$ and the value $u_{16,16}^m$ at the midpoint $(16h, 16h) = (\frac{1}{2}, \frac{1}{2})$ of Ω. The values $u_{16,16}^m$ should converge to

$$u\left(\frac{1}{2}, \frac{1}{2}\right) = 0.5.$$

The listed values indicate the convergence of the Gauss–Seidel method, but its slowness is disappointing. After 100 iterations the first decimal of $u_{16,16}^m$ is still completely wrong! The third column contains the so-called *reduction factor*: the ratio $\varepsilon_{m-1}/\varepsilon_m$ of the successive errors. This factor indicates how fast the error decreases per iteration. A comparison of the data in Table 1.1 demonstrates that the ordering influences the results, but not the convergence speed.

	lexicographical ordering			chequer-board ordering		
m	$u_{16,16}^m$	ε_m	$\varepsilon_{m-1}/\varepsilon_m$	$u_{16,16}^m$	ε_m	$\varepsilon_{m-1}/\varepsilon_m$
0	0.0	1.877	-	0.0	1.877	-
1	-0.002	1.760	0.93756	-0.001	1.759	0.93704
2	-0.004	1.646	0.93563	-0.003	1.589	0.90323
9	-0.018	1.276	-	-0.017	1.202	-
10	-0.019	1.246	0.97637	-0.019	1.165	0.96903
99	+0.1102	0.404	-	+0.1353	0.380	-
100	+0.1135	0.400	0.98989	+0.1385	0.376	0.98994
199	+0.3479	0.152	-	+0.3585	0.142	-
200	+0.3494	0.151	0.99041	+0.3598	0.140	0.99041
299	+0.4421	0.058	-	+0.4461	0.054	-
300	+0.4426	0.057	0.99039	+0.4466	0.053	0.99039

Table 1.1 Results of the Gauss–Seidel iteration for $N = 32$.

The Gauss–Seidel iteration (1.15) is equivalent to the representation

$$\textbf{for } i := 1 \textbf{ to } n \textbf{ do } x_i^{m+1} := x_i^m - \left[\sum_{j=1}^{i-1} a_{ij}x_j^{m+1} + \sum_{j=i}^{n} a_{ij}x_j^m - b_i\right] / a_{ii}, \tag{1.21}$$

showing that the new iterate x_i^{m+1} is obtained from x_i^m by subtracting a correction. In contrast to (1.15), the second sum in (1.21) starts at $j = i$. A seemingly insignificant modification is the multiplication of this correction by a factor ω. The obtained method is called the *successive overrelaxation method* and abbreviated with *SOR*. In the general case, it takes the form

m	$u_{16,16}^m$	ε_m	$\frac{\varepsilon_{m-1}}{\varepsilon_m}$	m	$u_{16,16}^m$	ε_m	$\frac{\varepsilon_{m-1}}{\varepsilon_m}$
0	0.0	1.877	-				
1	-0.016	1.777	0.9468	39	0.4805	0.050	-
2	-0.027	1.680	0.9451	40	0.4838	0.043	0.8566
9	-0.065	1.046	-	49	0.4964	0.0055	-
10	-0.068	0.962	0.9197	50	0.4970	0.0049	0.8830
19	0.1111	0.399	-	99	0.4999996	9.05_{10}-7	-
20	0.1486	0.365	0.9155	100	0.4999997	7.23_{10}-7	0.7977
29	0.4198	0.166	-	129	0.5-1.5_{10}-9	3.57_{10}-9	-
30	0.4445	0.150	0.9062	130	0.5-1.2_{10}-9	2.81_{10}-9	0.7881

Table 1.2 SOR (lexicographical ordering, $N = 32$, $\omega = 1.821465$).

$$\textbf{for } i := 1 \textbf{ to } n \textbf{ do } x_i^{m+1} := x_i^m - \frac{\omega}{a_{ii}} \left[\sum_{j=1}^{i-1} a_{ij} x_j^{m+1} + \sum_{j=i}^{n} a_{ij} x_j^m - b_i \right]. \tag{1.22}$$

In the model case, the only change in (1.19) or (1.20) is the replacement of the assignment $u[i,j] := (\dots)/4$ by

$$u[i,j] := u[i,j] - \tfrac{\omega}{4} \left[4u[i,j] - u[i-1,j] - u[i+1,j] - u[i,j-1] - u[i,j+1] - h^2 \cdot f[i,j]\right].$$

In §4.6 we shall prove that $\omega = 2/(1+\sin(\pi h))$ (i.e., $\omega = 1.821\dots$ for $N = 32$) is a suitable value. Table 1.2 shows the errors ε_m of the first 150 iterations for the same example as above. Convergence is evidently much faster than for the Gauss–Seidel method. Analysis of the above mentioned methods and constructing even faster iterations are the foci of the next chapters.

1.7 Sparse Matrices Versus Fully Populated Matrices

The matrix of the Poisson model problem is an example of a sparse matrix. Discretisation by finite differences and finite elements (cf. §E.2) generates sparse matrices with the property that the number of nonzero entries per row is bounded,[8] i.e., the number

$$s(I) := \max_{\alpha \in I} \#\{\beta \in I : a_{\alpha\beta} \neq 0\} \tag{1.23}$$

is bounded independently of the matrix size $n = \#I$. This property is important for the storage cost which is $\mathcal{O}(n)$. Operations as matrix-vector multiplication and most of the operations involved by one step of an iteration method also have a computational cost of $\mathcal{O}(n)$.

Formally, the iterative schemes also work for fully populated matrices. In this case, the storage cost is n^2 and basic operations as matrix-vector multiplication cost $\mathcal{O}(n^2)$ arithmetic operations. For large-scale matrices, the quadratic order $\mathcal{O}(n^2)$ is too large. Fortunately, there are other techniques which allow reducing $\mathcal{O}(n^2)$ to[9] $\mathcal{O}(n \log^* n)$ or even $\mathcal{O}(n)$. Appendix D will describe such a method.

There is a further reason why in the following we focus to sparse matrices. Fully populated matrices typically arise from discretising nonlocal operators (integral operators), e.g., by the boundary element method (cf. Sauter–Schwab [331]). The corresponding linear systems have other properties than, e.g., the Poisson model problem, and require other types of iterative methods (see the Picard iteration and the multigrid iteration of the second kind in §11.9.1).

Assume that A is a sparse matrix satisfying (1.23). In the regular case of a difference scheme, the data of A are organised by a nine-point star (1.13a) or a five-point formula (1.14). If the coefficients are constant as in the left-hand side of (1.13a) or

[8] For the finite element method, this is a consequence of the shape regularity of the triangulation.

[9] The asterix in $\log^* n$ indicates an unspecified power.

in (1.14), the data size is negligible. Procedure (1.19) shows how easily these data can be used.

However, the standard case are more general sparse matrices arising from finite element discretisations. In this case, the sparse matrix format is used for organising the matrix data. For each $i \in I$, there is a subset

$$I_\alpha := \{\beta \in I : a_{\alpha\beta} \neq 0\}$$

whose size is bounded by $s(I)$ (cf. (1.23)). The entries $a_{\alpha\beta} \neq 0$ (α fixed) form the vector $\mathbf{a}_\alpha \in \mathbb{K}^{I_\alpha}$. Then the matrix data are described by the set

$$\{(I_\alpha, \mathbf{a}_\alpha) : \alpha \in I\},$$

which can be implemented by a list. For instance, the SOR iteration (1.22) reads as

$$\textbf{for } \alpha \in I \textbf{ do } \; u[\alpha] := u[\alpha] - \frac{\omega}{\mathbf{a}_\alpha[\alpha]} \left[\sum_{\beta \in I_\alpha} \mathbf{a}_\alpha[\beta] \cdot u[\beta] - h^2 \cdot f[\alpha] \right],$$

where the loop follows the ordering of I.

Chapter 2
Iterative Methods

Abstract In this chapter we consider general properties of iterative methods. Such properties are *consistency*, ensuring the connection between the iterative method and the given system of equations, as well as *convergence*, guaranteeing the success of the iteration. The most important result of this chapter is the characterisation of the convergence of linear iterations by the spectral radius of the iteration matrix (cf. §2.1.4). Since we only consider iterative methods for systems with regular matrices, iterative methods for singular systems or those with rectangular matrices will not be studied.[1] The quality of a linear iteration depends on both the cost and the convergence speed. The resulting efficacy is discussed in Section 2.3. Finally, Section 2.4 explains how to test iterative methods numerically.

2.1 Consistency and Convergence

2.1.1 Notation

We want to solve the *system of linear equations*

$$Ax = b \qquad (A \in \mathbb{K}^{I\times I} \text{ and } b \in \mathbb{K}^{I} \text{ given}) \tag{2.1}$$

(cf. (1.10)). To guarantee solvability for all $b \in \mathbb{K}^I$, we generally assume:

$$A \text{ is regular.} \tag{2.2}$$

An iterative method producing iterates $x^1, x^2, \ldots$ from the starting value x^0 can be characterised by a prescription $x^{m+1} := \Phi(x^m)$. Φ depends on the data A and b in (2.1). These parameters are explicitly expressed by the notation

[1] Concerning this topic, we refer, e.g., to Björck [47], Marek [275], Kosmol–Zhou [241], Berman–Plemmons [46], and Remark 5.17.

W. Hackbusch, *Iterative Solution of Large Sparse Systems of Equations*, Applied Mathematical Sciences 95, DOI 10.1007/978-3-319-28483-5_2

$$x^{m+1} := \Phi(x^m, b, A) \qquad (m \geq 0,\ b \text{ in (2.1)}). \tag{2.3}$$

Since in most of the cases the matrix A is fixed, we usually write

$$x^{m+1} := \Phi(x^m, b)$$

instead of $\Phi(x^m, b, A)$. By $\Phi(\cdot, \cdot, A)$ we express the fact that we consider the iteration (2.3) exclusively for the matrix A.

Definition 2.1. An *iterative method* is a (in general nonlinear) mapping

$$\Phi : \mathbb{K}^I \times \mathbb{K}^I \times \mathbb{K}^{I\times I} \to \mathbb{K}^I.$$

By $x^m = x^m(x^0, b, A)$ we denote the *iterates* of the sequence generated by the prescription (2.3) with a starting value $x^0 = y \in \mathbb{K}^I$:

$$\begin{aligned} x^0(y, b, A) \quad &:= y\,, \\ x^{m+1}(y, b, A) &:= \Phi(x^m(y, b, A), b, A) \quad \text{for } m \geq 0. \end{aligned} \tag{2.4}$$

If A is fixed, we write $x^m(y, b)$ instead of $x^m(y, b, A)$. If all parameters y, b, A are fixed, we write x^m.

If Φ is called an *iteration method*, we expect that the method is applicable to a whole class of matrices A. Here ‘applicable’ means that Φ is well defined (including the case that the sequence x^m diverges).

Definition 2.2. (a) $\mathfrak{D}(\Phi) := \{A : \Phi(\cdot, \cdot, A) \text{ well defined}\}$ is the *domain of* Φ.
(b) An iteration is called *algebraic* if the definition of $\Phi(\cdot, \cdot, A)$ can be based exclusively on the data of $A \in \mathfrak{D}(\Phi)$.

In the case of the Gauss–Seidel iteration Φ^{GS} in (1.15), the domain is defined by $\mathfrak{D}(\Phi^{\mathrm{GS}}) = \{A \in \mathbb{K}^{I\times I} : a_{ii} \neq 0 \text{ for all } i \in I,\ I \text{ finite}\}$. Another extreme case is $\mathfrak{D}(\Phi) = \{A\}$, i.e., the iteration can only be applied to one particular matrix A.

2.1.2 Fixed Points

Definition 2.3. $x^* = x^*(b, A)$ is called a fixed point of the iteration Φ corresponding to $b \in \mathbb{K}^I$ and $A \in \mathfrak{D}(\Phi)$ (or shortly: a fixed point of $\Phi(\cdot, b, A)$) if

$$x^* = \Phi(x^*, b, A).$$

If the sequence $\{x^m\}$ of the iterates generated by (2.3) converges, we may form the limit in (2.3) and obtain the next lemma.

Lemma 2.4. *Let the iteration Φ be continuous with respect to the first argument. If*

$$x^* := \lim_{m\to\infty} x^m(y, b, A) \qquad \textit{(cf. (2.4))}$$

exists, x^* is a fixed point of $\Phi(\cdot, b, A)$.

2.1.3 Consistency

Lemma 2.4 states that possible results of the iteration method have to be sought in the set of fixed points. Therefore, a minimum condition is that the solution of system (2.1) with the right-hand side $b \in \mathbb{K}^I$ be a fixed point with respect to b. This property is the subject of the following definition.

Definition 2.5 (consistency). The iterative method $\varPhi$ is called *consistent* to the system (2.1) with $A \in \mathfrak{D}(\varPhi)$ if, for all right-hand sides $b \in \mathbb{K}^I$, any solution of $Ax = b$ is a fixed point of $\varPhi(\cdot, b, A)$.

According to Definition 2.5, consistency means: For all $b, x \in \mathbb{K}^I$ and all matrices $A \in \mathfrak{D}(\varPhi)$, the implication $Ax = b \Rightarrow x = \varPhi(x, b, A)$ holds. The reverse implication would yield an *alternative* (nonequivalent) form of consistency:

$$Ax = b \quad \text{for all fixed points } x \text{ of } \varPhi(\cdot, b, A) \text{ and for all } b \in \mathbb{K}^I,\ A \in \mathfrak{D}(\varPhi). \tag{2.5}$$

Note that both variants of consistency do not require the regularity assumption (2.2). Even without (2.2), there may be a solution of $Ax = b$ for certain b. Then Definition 2.5 implies the existence of a fixed point of $\varPhi(\cdot, b)$. Vice versa, (2.5) states the existence of a solution of $Ax = b$ as soon as $\varPhi(\cdot, b, A)$ has a fixed point. The regularity of A will be discussed in Theorem 2.8.

2.1.4 Convergence

A natural definition of the convergence of an iterative method $\varPhi$ seems to be

$$\lim_{m\to\infty} x^m(y, b, A) \quad \text{exists for all } y, b \in \mathbb{K}^I, \tag{2.6}$$

where $x^m(y, b, A)$ are the iterates defined in (2.4) corresponding to the starting value $x^0 := y$, while $A \in \mathfrak{D}(\varPhi)$ is a fixed matrix. Since the starting value may be chosen arbitrarily, it may happen that an iteration satisfying (2.6) converges, but to different limits *depending* on the starting value. Therefore, the independence of the limit has to be incorporated into the definition of convergence. This yields the following definition, which is stronger than (2.6).

Definition 2.6. Fix $A \in \mathfrak{D}(\varPhi)$. An iterative method $\varPhi(\cdot, \cdot, A)$ is called *convergent* if for all $b \in \mathbb{K}^I$, there is a limit $x^*(b, A)$ of the iterates (2.4) independent of the starting value $x^0 = y \in \mathbb{K}^I$.

Note that consistency is a property of $\varPhi$ for *all* $A \in \mathfrak{D}(\varPhi)$, whereas convergence is required for a particular $A \in \mathfrak{D}(\varPhi)$. Therefore $\varPhi(\cdot, \cdot, A)$ may be convergent for some A, while $\varPhi(\cdot, \cdot, A')$ diverges for another A'.

2.1.5 Convergence and Consistency

Remark 2.7. In the following, we shall often assume that the iterative method Φ is *convergent and consistent*. The term 'convergent and consistent' refers to a matrix $A \in \mathfrak{D}(\Phi)$ and means precisely: Φ is consistent and, for $A \in \mathfrak{D}(\Phi)$, the particular iteration $\Phi(\cdot, \cdot, A)$ is convergent.

It will turn out that the chosen definitions of the terms 'convergence' and 'consistency' of Φ are almost equivalent to the combination of the alternative definitions in (2.5) and (2.6).

Theorem 2.8. *Let Φ be continuous in the first argument. Then Φ is consistent and convergent if and only if A is regular and Φ fulfils the conditions (2.5) and (2.6).*

Proof. (i) Assume Φ to be consistent and convergent. (2.6) follows from Definition 2.6. If A is singular, the equation $Ax = 0$ would have a nontrivial solution $x^{**} \neq 0$ besides $x^* = 0$. By consistency, both are fixed points of Φ with respect to $b = 0$. Therefore, choosing the starting values $x^0 = x^*$ and $x^0 = x^{**}$, we obtain the constant sequences $x^m(x^*, 0) = x^*$ and $x^m(x^{**}, 0) = x^{**}$. The convergence definition states that the limits x^* and x^{**} coincide contrary to the assumption. Hence, A is regular. It remains to prove (2.5). The preceding argument shows that a convergent iterative method can have only *one* fixed point with respect to b. Because of the regularity of A, there is a solution of $Ax = b$ that, thanks to consistency, is the unique fixed point of Φ with b. Hence, (2.5) is proved.

(ii) Assume $\Phi(x, b)$ to be continuous in x and that (2.5) and (2.6) are fulfilled. Furthermore, let A be regular. Due to Lemma 2.4, $x^* := \lim x^m(y, b)$ is a fixed point of Φ with respect to b and therefore, by (2.5), a solution of $Ax = b$. Because of the regularity of A, the solution of the system is unique and hence also the limit of $x^m(y, b)$, which thereby cannot depend on y. Hence, Φ is convergent in the sense of Definition 2.6. Convergence leads to the uniqueness of the fixed point with respect to b (cf. part (i)). Since, by (2.5), this fixed point is the uniquely determined solution of $Ax = b$, Φ is consistent. □

2.1.6 Defect Correction as an Example of an Inconsistent Iteration

In this monograph, all iterations will be assumed to be consistent. Usually, inconsistent iterations are an involuntary consequence of a bug in the implementation. However, there are examples where inconsistent iterations are of practical relevance. Assume that both $Ax = b$ and $By = c$ are discretisations of the same partial differential equation. Assume further that $Ax = b$ is simpler to solve than $By = c$, but the error of the discretisation by B is smaller than the discretisation error of A. Then there are combinations of both discretisations so that the overall treatment is as simple as for A but yielding the accuracy of B.

The standard defect correction $x^{m+1} = x^m - A^{-1}(Bx^m - c)$ can be stopped after a few iteration steps since the desired discretisation accuracy is reached (cf. [194, §14.2.2], [197, §7.5.9.2]). This is even true if the matrix B is singular or almost singular (this is the case of an unstable but consistent[2] discretisation). An extreme case of solving a problem with an unstable discretisation of high consistency order is demonstrated in [178].

Another mixing of both discretisation is described in [194, §14.3.3], where parts of the multigrid iteration for $Ax = b$ use B in the smoothing step. The limit x^* of the iterates solves neither $Ax^* = b$ nor $Bx^* = c$.

2.2 Linear Iterative Methods

One would expect iterative methods to be linear in x, b, since they solve linear equations. In fact, most of the methods described in this book are linear, but there are also important nonlinear iterations as, e.g., discussed in Part II.

2.2.1 Notation, First Normal Form

Definition 2.9 (linear iteration, iteration matrix). An iterative method Φ is called *linear* if $\Phi(x, b)$ is linear in (x, b), i.e., if there are matrices M and N such that

$$\Phi(x, b, A) = M[A]\,x + N[A]\,b.$$

In most of the cases, A is fixed and we use the shorter form

$$\Phi(x, b) = Mx + Nb. \tag{2.7}$$

Here, the matrix $M = M[A]$ is called the *iteration matrix* of the iteration Φ.

Iteration (2.3) takes the form (2.8), which represents the *first normal form* of the iteration Φ:

$$x^{m+1} := Mx^m + Nb \qquad (m \geq 0,\ b \text{ in (2.1)}). \tag{2.8}$$

Whenever possible, we shall denote the iteration matrix of a specific iteration method 'xyz' by M^{xyz}; e.g., M^{GS} belongs to the Gauss-Seidel method. Similarly for N^{xyz}. When we refer to the mapping Φ, we write M_Φ, N_Φ, etc.

Remark 2.10. Assume (2.2). If $N = N[A]$ is singular, there is some $x^* \neq 0$ with $Nx^* = 0$ and $b := Ax^* \neq 0$. Starting iteration (2.8) with $x^0 = 0$ yields $x^m = 0$ and hence $\lim x^m = 0$. In Corollary 2.17b we shall state that, in this case, the iteration is not convergent.

The iteration $\Phi(\cdot, \cdot, A)$ is algebraic in the sense of Definition 2.2b if and only if the matrices M and N are explicit functions of A.

[2] Concerning the terms 'consistent' and 'consistency order', we refer to Hackbusch [197, §§6,7].

2.2.2 Consistency and Second Normal Form

For a linear and consistent iteration Φ, each solution of $Ax = b$ must be a fixed point with respect to b: $x = Mx + Nb$. Each $x \in \mathbb{K}^I$ can be the solution of $Ax = b$ (namely, for $b := Ax$). Hence,

$$x = Mx + Nb = Mx + NAx$$

holds for all x and leads to the matrix equation

$$M[A] + N[A]A = I, \tag{2.9}$$

or in short,

$$M + NA = I,$$

establishing a relation between M and N in (2.8). This proves the next theorem.

Theorem 2.11 (consistency). *A linear iteration Φ is consistent if and only if the iteration matrix M can be determined from N by*

$$M[A] = I - N[A]\,A \qquad \textit{for all } A \in \mathfrak{D}(\Phi). \tag{2.9$'$}$$

If, in addition, A is regular, N can be represented as a function of M:

$$N[A] = (I - M[A])A^{-1}. \tag{2.9$''$}$$

Combining formulae (2.8) and (2.9′), we can represent linear and consistent iterations in their *second normal form*:

$$x^{m+1} := x^m - N[A]\,(Ax^m - b) \qquad (m > 0,\ A, b \text{ in } (2.1)). \tag{2.10}$$

In the sequel, the matrix

$$N = N[A] = N_\Phi = N_\Phi[A]$$

will be called the ‘matrix of the second normal form of Φ’. Equation (2.10) shows that x^{m+1} is obtained from x^m by a correction which is the *defect* $Ax^m - b$ of x^m multiplied by N. The fact that the defect of x^m vanishes if and only if it is a solution of $Ax = b$, proves the next remark.

Remark 2.12. The second normal form (2.10) with arbitrary $N \in \mathbb{K}^{I\times I}$ represents all linear and consistent iterations.

Since consistent linear iterations are the standard case, we introduce the following notation for the set of these iterations:

$$\mathcal{L} := \{\Phi : \mathbb{K}^I \times \mathbb{K}^I \times \mathbb{K}^{I\times I} \to \mathbb{K}^I \text{ consistent linear iteration}, \#I < \infty\}. \tag{2.11}$$

2.2.3 Third Normal Form

The *third normal form* of a linear iteration reads as follows:

$$W[A]\,(x^m - x^{m+1}) = Ax^m - b \qquad (m > 0,\ A, b \text{ in } (2.1)). \tag{2.12}$$

$W = W[A] = W_\Phi = W_\Phi[A]$ is called the 'matrix of the third normal form of Φ'. Equation (2.12) can be understood in the following algorithmic form:

$$\text{solve } W\delta = Ax^m - b \qquad \text{and define} \qquad x^{m+1} := x^m - \delta. \tag{2.12$'$}$$

This represents a definition of x^{m+1} as long as W is regular. Under this assumption, one can solve for x^{m+1}. A comparison with (2.10) proves the following.

Remark 2.13. If W in (2.12) is regular, iteration (2.12) coincides with the second normal form (2.10), where N is defined by

$$N = W^{-1}. \tag{2.13}$$

Vice versa, the representation (2.10) with regular N can be rewritten as (2.12) with $W = N^{-1}$.

We shall see that for the interesting cases, N must be regular (cf. Remark 2.18). Combining (2.9$'$) and (2.13) yields

$$M[A] = I - W[A]^{-1}A. \tag{2.13$'$}$$

2.2.4 Representation of the Iterates x^m

By the notation $x^m(x^0, b, A)$ in (2.4) we express the dependency on the starting value x^0 and on the the data b, A of the system (2.1). The explicit representation of x^m in terms of x^0 and b is given in (2.14).

Theorem 2.14. *The linear iteration (2.7) produces the iterates*

$$x^m(x^0, b, A) = M[A]^m x^0 + \sum_{k=0}^{m-1} M[A]^k N[A]\, b \tag{2.14}$$

for $m \geq 0$ *and* $A \in \mathfrak{D}(\Phi)$.

Proof. For the induction start at $m = 0$, Eq. (2.14) takes the form $x^0(x^0, b) = x^0$ in accordance with (2.4). Assuming (2.14) for $m - 1$, we obtain from (2.7) that

$$\begin{aligned} x^m(x^0, b) &= Mx^{m-1} + Nb = M\left(M^{m-1}x^0 + \sum_{k=0}^{m-2} M^k Nb\right) + Nb \\ &= M^m x^0 + \sum_{k=1}^{m-1} M^k Nb + Nb = M^m x^0 + \sum_{k=0}^{m-1} M^k Nb. \end{aligned}$$

□

In the following, e^m denotes the (iteration) error of x^m:

$$e^m := x^m - x, \qquad \text{where } x \text{ solves } Ax = b. \tag{2.15}$$

Assuming consistency, we have $x = Mx + Nb$ for the solution x in (2.15). Forming the difference with (2.8): $x^{m+1} = Mx^m + Nb$, we attain the simple relation

$$e^{m+1} = Me^m \quad (m \geq 0), \qquad e^0 = x^0 - x, \tag{2.16a}$$

between two successive errors. Therefore the iteration matrix is the amplification matrix of the error. A trivial conclusion is

$$e^m = M^m e^0 \qquad (m \geq 0). \tag{2.16b}$$

The expression $Ax - b$ is called the *defect* of a vector x. In particular,

$$d^m := Ax^m - b \tag{2.17}$$

denotes the defect of the m-th iterate x^m.

Exercise 2.15. Prove: (a) The defect $\bar{d} = A\bar{x} - b$ and the error $\bar{e} = \bar{x} - x$ fulfil the equation $A\,\bar{e} = \bar{d}$.
(b) Let $\Phi \in \mathcal{L}$ (cf. (2.11)) and assume that A is regular. Then the defects satisfy

$$d^{m+1} = AMA^{-1}d^m, \quad d^0 := Ax^0 - b, \quad d^m = (AMA^{-1})^m d^0.$$

2.2.5 Convergence

A necessary and sufficient convergence criterion can be formulated by the spectral radius $\rho(M)$ of the iteration matrix (cf. Definition A.17).

Theorem 2.16 (convergence theorem, convergence rate). *A linear iteration (2.7) with the iteration matrix $M = M[A]$ is convergent if and only if*

$$\rho(M) < 1. \tag{2.18}$$

$\rho(M)$ *is called the* convergence rate *of the iteration* $\Phi(\cdot,\cdot,A)$.

In the sequel, the terms *convergence rate*, *convergence speed*, and *iteration speed* are used synonymously for $\rho(M)$. Some authors define the convergence rate as the negative logarithm $-\log(\rho(M))$ (cf. (2.30a) and Varga [375], Young [412]).

Proof. (i) Let iteration (2.7) be convergent. In Definition 2.6 we may choose $b := 0$ and exploit the representation (2.14): $x^m = M^m x^0$. The starting value $x^0 := 0$ yields the limit $x^* = 0$, which by the convergence definition must hold for any starting value. If $\rho(M) \geq 1$, one could choose $x^0 \neq 0$ as the eigenvector corresponding to an eigenvalue λ with $|\lambda| = \rho(M) \geq 1$. The resulting sequence $x^m = \lambda^m x^0$ cannot converge to $x^* = 0$. Hence, inequality (2.18) is necessary for convergence.

(ii) Now let (2.18) be valid: $\rho(M) < 1$. By Lemma B.28, $M^m x^0$ converges to zero, while Theorem B.29 proves $\sum_{k=0}^{m-1} M^k \to (I - M)^{-1}$. Thanks to the representation (2.14), x^m tends to

$$x^* := (I - M)^{-1} N b. \tag{2.19}$$

Since this limit does not depend on the starting value, the iteration is convergent. □

The proof already contains the first statement of the following corollary.

Corollary 2.17. (a) If the iterative method (2.7) is convergent, the iterates converge to $(I - M)^{-1} N b$.
(b) If the iteration is convergent, then A and $N = N[A]$ are regular.
(c) If, in addition, the iteration is consistent, the iterates x^m converge to the unique solution $x = A^{-1} b$.

Proof. (b) If either A or N are singular, the product AN is singular and $ANx = 0$ holds for some $x \neq 0$. As $M = I - NA$, x is an eigenvector of M with the eigenvalue 1. Hence $\rho(M) \geq 1$ proves the divergence of the iteration. This proves part (b).

(c) By consistency and part (b), there is a representation (2.10) with regular N and A, so that $(I - M)^{-1} N = A^{-1}$ follows from (2.9). (2.19) proves part (c). □

Remark 2.18. Since only convergent and consistent iterations are of interest and since in this case, by Corollary 2.17b, A and N are regular, the representation (2.9$''$) of N and the third normal form (2.4) hold with the matrix $W = N^{-1}$.

The convergence $x^m \to x$ is an asymptotic statement for $m \to \infty$ that allows no conclusion concerning the error $e^m = x^m - x$ for some fixed m. The values of $u^m_{16,16}$ given in Tables 1.1–1.2 even deteriorate during the first steps before they converge monotonically to the limit $\frac{1}{2}$. Often, one would like to have a statement for a *fixed* iteration number m. In this case, the convergence criterion (2.18) has to be replaced with a norm estimate.

Theorem 2.19. *Let* $\|\cdot\|$ *be a corresponding matrix norm. A sufficient condition for convergence of an iteration is the estimate*

$$\|M\| < 1 \tag{2.20}$$

of the iteration matrix M. *If the iteration is consistent, the error estimates (2.21) hold:*

$$\|e^{m+1}\| \leq \|M\| \, \|e^m\|, \qquad \|e^m\| \leq \|M\|^m \, \|e^0\|. \tag{2.21}$$

Proof. (2.20) implies (2.18) (cf. (B.20b)). (2.21) is a consequence of (2.16a,b). □

$\|M\|$ is called the *contraction number* of the iteration (with respect to the norm $\|\cdot\|$). In the case of (2.20), the iteration is called *monotonically convergent* with respect to the norm $\|\cdot\|$, since $\|e^{m+1}\| < \|e^m\|$. If the norm $\|\cdot\|$ fulfils the equality $\rho(M) = \|M\|$, the terms ‘convergence’ and ‘monotone convergence’ coincide.

2.2.6 Convergence Speed

Inequality (2.21), i.e., $\|e^{m+1}\| \le \zeta \, \|e^m\|$ with $\zeta := \|M\| < 1$, describes linear convergence. Faster convergence than linear convergence is only attainable by nonlinear methods (cf. §10.2.3). The contraction number ζ depends on the choice of the norm. According to (B.20b), the contraction number ζ is always larger or equal to the convergence rate $\rho(M)$. On the other hand, Lemma B.26 ensures that for a suitable choice of the norm, the contraction number ζ approximates the convergence rate $\rho(M)$ arbitrarily well.

The contraction number as well as the convergence rate determine the quality of an iterative method. Both quantities can be determined from the errors e^m as follows.

Remark 2.20. The contraction number is the maximum of the ratios $\|e^1\|/\|e^0\|$ taken over all starting values x.

Proof. Use (2.16b) for $m = 1$ and Exercise B.10d. □

Exercise 2.21. Prove: (a) In general, Remark 2.20 becomes wrong if $\|e^1\|/\|e^0\|$ is replaced with $\|e^{m+1}\| / \, \|e^m\|$ for some $m > 0$.
(b) The latter quotient takes the maximum

$$\zeta_{m+1} := \begin{cases} \max\{\|Mx\| \, / \, \|x\| : x \in \operatorname{range}(M^m)\backslash\{0\}\} & \text{if } M^m \ne 0, \\ 0 & \text{otherwise,} \end{cases}$$

which can be interpreted as the matrix norm of the mapping $x \mapsto Mx$ restricted to the subspace $V_m := \operatorname{range}(M^m) := \{M^m x : x \in \mathbb{K}^I\}$.
(c) The inclusion $V_{m+1} \subset V_m$ holds with an equality sign at least for $m \ge \#I$.
(d) $\rho(M) \le \zeta_{m+1} \le \zeta_m \le \zeta_0 = \zeta := \|M\|$ holds for $m \ge 0$.
(e) For regular M, one has $\zeta_m = \zeta$ for all m.

Exercise 2.21 demonstrates that the contraction number is a somewhat too coarse term: It may happen that the contraction number gives a too pessimistic prediction of the convergence speed. A more favourable estimate can be obtained by the numerical radius $r(\cdot)$ of the matrix M^m (cf. §B.3.4). The inequalities

$$\|M^m\|_2 \le 2\, r(M^m) \qquad \text{(cf. (B.28d))} \tag{2.22a}$$

and (2.16b) yield the error estimate

$$\|e^m\|_2 \le 2\, r(M^m)\, \|e^0\|_2 \qquad (m \ge 0) \tag{2.22b}$$

with respect to the Euclidean norm. If $\|\cdot\|_C$ is the norm defined by (C.5a) with a positive definite matrix C, one analogously proves the inequality

$$\|e^m\|_C \le 2\, r(C^{1/2} M^m C^{-1/2}) \|e^0\|_C \qquad (m \ge 0). \tag{2.22c}$$

For the practical judgment of the convergence speed from 'experimental data', i.e., from a sequence of errors e^m belonging to a special starting value x^0, one may use the *reduction factors*

$$\rho_{m+1,m} := \|e^{m+1}\| / \|e^{m}\| . \tag{2.23a}$$

These numbers can, e.g., be found in the last column of Tables 1.1–1.2. More interesting than a single value $\rho_{m+1,m}$ is the geometric mean

$$\rho_{m+k,m} := \left[\rho_{m+k,m+k-1} \cdot \rho_{m+k-1,m+k-2} \cdot \ldots \cdot \rho_{m+1,m}\right]^{1/k},$$

which due to definition (2.23a) can more easily be represented by

$$\rho_{m+k,m} := \left[\|e^{m+k}\| / \|e^{m}\|\right]^{1/k}. \tag{2.23b}$$

The properties of $\rho_{m+k,m}$ are summarised below.

Remark 2.22. (a) Denote the dependence of the magnitude $\rho_{m+k,m}$ on the starting value x^0 by $\rho_{m+k,m}(x^0)$. Then

$$\lim_{k\to\infty} \max\{\rho_{m+k,m}(x^0) : x^0 \in \mathbb{K}^I\} = \rho(M) \qquad \text{for all } m.$$

(b) Even without maximisation over all $x^0 \in \mathbb{K}^I$,

$$\lim_{k\to\infty} \rho_{m+k,m}(x^0) = \rho(M) \qquad \text{for all } m \tag{2.23c}$$

holds, provided that x^0 does not lie in the subspace $U \subset \mathbb{K}^I$ of dimension $< \#I$ spanned by all eigenvectors and possibly existing principal vectors of the matrix M corresponding to eigenvalues λ with $|\lambda| < \rho(M)$. (2.23c) holds almost always because a stochastically chosen starting value x^0 lying in a fixed lower dimensional subspace has probability zero.
(c) The reduction factors $\rho_{m+1,m}(x^0)$ tend to the spectral radius of M:

$$\lim_{m\to\infty} \rho_{m+1,m}(x^0) = \rho(M) \tag{2.23d}$$

for all $x^0 \notin U$ with U in part (b) if and only if there is exactly one eigenvalue $\lambda \in \sigma(M)$ with $|\lambda| = \rho(M)$, and if, for this eigenvalue, the geometric and algebraic multiplicities coincide. Sufficient conditions are: (i) $\lambda \in \sigma(M)$ with $|\lambda| = \rho(M)$ is a single eigenvalue, or (ii) M is a positive matrix (cf. (C.11a)).
(d) Choose a norm $\|\cdot\| = \|\cdot\|_C$ with $C > 0$ (cf. (2.22c)) in (2.23a). If $C^{\frac{1}{2}} M C^{-\frac{1}{2}}$ is Hermitian, $\rho_{m+1,m}(x^0)$ ($x^0 \notin U$) converges monotonically increasing to $\rho(M)$.

Proof. (i) Use

$$\rho(M) \le \max_{x^0\in\mathbb{K}^I} \rho_{m+k,m}(x^0) \le \max_{x^0\in\mathbb{K}^I} \rho_{k,0}(x^0) \le \|M^k\|^{1/k}$$

and $\|M^k\|^{1/k} \to \rho(M)$ according to Theorem B.27. This proves part (a).

(ii) Let $I_0 \subset I$ be the nonempty index subset $I_0 := \{i \in I : |J_{ii}| = \rho(M)\}$, where J_{ii} are the diagonal elements of the Jordan normal form $M = TJT^{-1}$ (cf. (A.15a,b)). The subspace $U := \{x : (T^{-1}x)_i = 0 \text{ for all } i \in I_0\}$ is the maximal subspace with the property $\lim_{m\to\infty} \left[\,\|M^m x\| \,/\, \|x\|\,\right]^{1/m} < \rho(M)$. Its dimension is $\dim(U) = \#I - \#I_0 < \#I$.

(iii) Define $\hat{M} = C^{1/2}MC^{-1/2}$ and $\hat{e}^m := C^{1/2}e^m$. Since the norms are related by $\|e^m\|_C = \|\hat{e}^m\|_2$, we obtain for $m \geq 1$ that

$$\begin{aligned}\|\hat{e}^m\|_2^2 &= \|\hat{M}^m\hat{e}^0\|_2^2 = \left\langle \hat{M}^m\hat{e}^0, \hat{M}^m\hat{e}^0 \right\rangle = \left\langle \hat{M}^{m+1}\hat{e}^0, \hat{M}^{m-1}\hat{e}^0 \right\rangle \\ &= \left\langle \hat{e}^{m+1}, \hat{e}^{m-1} \right\rangle \leq \|\hat{e}^{m+1}\|_2\|\hat{e}^{m-1}\|_2.\end{aligned}$$

Hence it follows that $\rho_{m+1,m} = \frac{\|e^{m+1}\|}{\|e^m\|} = \frac{\|\hat{e}^{m+1}\|_2}{\|\hat{e}^m\|_2} \geq \frac{\|\hat{e}^m\|_2}{\|\hat{e}^{m-1}\|_2} = \rho_{m,m-1}$. □

Remark 2.22 allows us to view the value $\rho_{m+k,m}$ and possibly also $\rho_{m+1,m}$ for sufficiently large m as a good approximation of the spectral radius. This viewpoint can be reversed.

Remark 2.23. The convergence rate $\rho(M)$ is a suitable measure for judging (asymptotically) the convergence speed. This holds even if convergence is required with respect to a specific norm.

Proof. By Theorem B.27, for each $\varepsilon > 0$ there is some m_0 such that $m \geq m_0$ implies that $\rho(M) \leq \|M^m\|^{1/m} \leq \rho(M) + \varepsilon$ and $\|e^m\| \leq (\rho(M) + \varepsilon)^m\|e^0\|$. □

2.2.7 Remarks Concerning the Matrices M, N, and W

Considerations in §§2.2.5–2.2.6 show the close connection between the iteration matrix M and the convergence speed. M directly describes the *error reduction* or amplification (cf. (2.16a)). Roughly speaking, the convergence is better the smaller M is. $M = 0$ would be optimal. However, then Φ is a direct method, since x^1 is already the exact solution (its error is $e^1 = Me^0 = 0$).

The matrix N transforms the defect $Ax^m - b$ into the correction $x^m - x^{m+1}$. The optimal case[3] $M = 0$ mentioned above corresponds to $N[A] = A^{-1}$. Therefore, one may regard $N[A]$ as an *approximate inverse* of A.

Concerning implementation, often the matrix W of the third normal form (2.12) is the important one. By the relation $W = N^{-1}$ (cf. (2.13)), $W = A$ would be optimal. However, then computing the correction $x^m - x^{m+1}$ is equivalent to the direct solution of the original equation. Therefore, one has to find approximations W of A, so that the solution of the system $W\delta = d$ is sufficiently easy.

In the case of some of the classical iterations discussed in §3, we have explicit expressions for N or W and may use these matrices for the computation. On the other hand, there will be iterative methods, for which the algorithm is implemented differently without reference to the matrices M, N, W (see, e.g., Propositions 3.13 or 5.25).

[3] Consistent linear iterations with $M = 0$ can be called direct solvers. Vice versa, any direct solver defines a linear iteration with $M = 0$.

2.2.8 Three-Term Recursions, Two- and Multi-Step Iterations

So far we considered *one-step iterations*, i.e., x^{m+1} is computed in one step from x^m. Sometimes linear iterations occur, in which computing x^{m+1} involves x^m and x^{m-1}:

$$x^{m+1} = M_0\, x^m + M_1\, x^{m-1} + N_0\, b \qquad (m \geq 1). \tag{2.24}$$

For the starting procedure, one needs two initial values x^0 and x^1. Such *two-step iterations* are also called *three-term recursions* since they involved the three terms x^{m+1}, x^m, x^{m-1}. Formally, a three-term recursion can be reduced to a standard one-step iteration acting in the space $\mathbb{K}^I \times \mathbb{K}^I$:

$$\begin{bmatrix} x^{m+1} \\ x^m \end{bmatrix} = \mathbf{M} \begin{bmatrix} x^m \\ x^{m-1} \end{bmatrix} + \begin{bmatrix} N_0 b \\ 0 \end{bmatrix} \qquad \text{with } \mathbf{M} := \begin{bmatrix} M_0 & M_1 \\ I & 0 \end{bmatrix}. \tag{2.25}$$

Now the convergence condition

$$\rho(\mathbf{M}) < 1 \tag{2.26a}$$

ensures that recursion (2.25) has a limit that is also the fixed point. The consistency condition takes the form

$$I - M_0 - M_1 = N_0 A\,. \tag{2.26b}$$

Exercise 2.24. The limit of the iteration (2.25) has the general form $\left[\begin{smallmatrix} \xi \\ \eta \end{smallmatrix}\right] \in \mathbb{K}^I \times \mathbb{K}^I$. Show that the conditions (2.26a,b) imply $\xi = \eta = A^{-1} b$.

Exercise 2.25. Given an iteration $x^{m+1} = Mx^m + Nb$, define the matrices M_0, M_1, N_0 in (2.24) by

$$\begin{aligned} M_0 &:= \Theta M + \vartheta I, \\ M_1 &:= (1 - \Theta - \vartheta)\, I, \\ N_0 &:= \Theta N \end{aligned}$$

with $\Theta, \vartheta \in \mathbb{R}$. The three-term recursion (2.24) takes the form

$$x^{m+1} = \Theta \left[(Mx^m + Nb) - x^{m-1} \right] + \vartheta (x^m - x^{m-1}) + x^{m-1}. \tag{2.27}$$

Prove that (a) $\mathbf{M}$ has the spectrum

$$\sigma(\mathbf{M}) = \left\{ \frac{1}{2} (\Theta\lambda + \vartheta) \pm \sqrt{1 - \Theta - \vartheta + \frac{1}{4} (\Theta\lambda + \vartheta)^2} \;:\; \lambda \in \sigma(M) \right\}.$$

(b) Conclude from $\rho(M) < 1$ and $\Theta > 0,\ \vartheta \geq 0,\ \Theta + \vartheta \leq 1$ that $\rho(\mathbf{M}) < 1$.

2.3 Efficacy of Iterative Methods

The convergence rate cannot be the only criterion for the quality of an iterative method because one has also to take into account the amount of computational work of Φ.

2.3.1 Amount of Computational Work

The representation (2.12′) suggests that any iteration requires at least computing the defect $Ax^m - b$. For a general $n \times n$ matrix $A \in \mathbb{K}^{I \times I}$ $(n = \#I)$, multiplying Ax^m would require $2n^2$ operations. However, as discussed in §1.7, it is more realistic to assume that A is sparse; i.e., the number $s(n)$ of the nonzero elements of A is distinctly smaller than n^2. For matrices arising from discretisations of partial differential equations, one has

$$s(n) \leq C_A n, \tag{2.28}$$

where C_A is a constant with respect to n, but depends on the matrix A. For the five-point formula (1.4a) of the model problem, inequality (2.28) holds with $C_A = 5$. Under assumption (2.28), one can perform matrix-vector multiplication in $2C_A n$ operations.

After evaluating $d := Ax^m - b$, one has still to solve the system $W\delta = d$ in (2.12′). For any practical iterative method, we should require that this part consumes only $\mathcal{O}(n)$ operations, so that the total amount of work is also of the order $\mathcal{O}(n)$. We relate the constant in $\mathcal{O}(n)$ to C_A in (2.28) and obtain the following formulation:

The number of arithmetic operations per iteration step of the method Φ is $\mathrm{Work}\,(\Phi, A) \leq C_\Phi C_A n\,.$ (2.29)

Here, $\mathrm{Work}\,(\Phi, A)$ is the amount of work of the Φ iteration applied to $Ax = b$. Note that C_Φ depends on the iteration Φ but not on A, whereas $C_A n$ indicates the degree of sparsity of A. Therefore, the constant C_Φ may be called the *cost factor* of the iteration Φ.

So far we only discussed the cost arising by performing one iteration step of Φ. Depending on the method, some *initialisation* may be necessary for precomputing some quantities required by Φ. Let $\mathrm{Init}(\Phi, A)$ be the corresponding cost.

Remark 2.26. If m iteration steps are performed, the effective cost per iteration is

$$\mathrm{Work}\,(\Phi, A) + \mathrm{Init}(\Phi, A)/m.$$

In the standard case, the initialisation uses only the data of A. Therefore it pays if many systems $Ax^i = b^i$ are solved with different right-hand sides b^i but the same matrix A.

2.3.2 Efficacy

An iteration Φ can be called 'more effective' than Ψ if for the same amount of work Φ is faster, or if Φ has the same convergence rate, but consumes less work than Ψ. To obtain a common measure, we ask for the amount of work that is necessary to reduce the error by a fixed factor. This factor is chosen as $1/\mathrm{e}$, since the natural logarithm is involved. According to Remark 2.23, we use the convergence rate $\rho(M)$ for the (asymptotic) description of the error reduction per iteration step. After m iteration steps, the asymptotic error reduction is $\rho(M)^m$. In order to ensure $\rho(M)^m \le 1/\mathrm{e}$, we have to choose $m \ge -1/\log(\rho(M))$, provided that convergence holds: $\rho(M) < 1 \Leftrightarrow \log(\rho(M)) < 0$. Therefore, we define

$$\mathrm{It}(\Phi) := -1/\log(\rho(M)). \tag{2.30a}$$

$\mathrm{It}(\Phi)$ represents the (asymptotic) number of the iteration steps for an error reduction by the factor of $1/\mathrm{e}$. Note that, in general, $\mathrm{It}(\Phi)$ is not an integer.

Remark 2.27. (a) Convergence of Φ is equivalent to $0 \le \mathrm{It}(\Phi) < \infty$. The value $\mathrm{It}(\Phi) = 0$ corresponds to $\rho(M) = 0$, i.e., to a direct method.

(b) Let $\Phi \in \mathcal{L}$. To reduce the iteration error (asymptotically) by a factor of $\varepsilon < 1$, we need the following number of iteration steps:

$$\mathrm{It}(\Phi, \varepsilon) := -\mathrm{It}(\Phi)\log(\varepsilon) \tag{2.30b}$$

(c) If $\rho(M) = \|M\|$ or $\rho(M)$ in (2.30a) is replaced with $\|M\| < 1$, one can guarantee (not only asymptotically) that

$$\|e^{m+k}\| \le \varepsilon \, \|e^m\| \qquad \text{for } k \ge \mathrm{It}(\Phi, \varepsilon). \tag{2.30c}$$

(d) If $r(M) < 1$ holds for the numerical radius of M introduced in §B.3.4, definition (2.30b) can be replaced with $\mathrm{It}(\Phi, \varepsilon) := \log(\varepsilon/2)/\log(r(M))$. Then, inequality (2.30c) holds with respect to the Euclidean norm.

The amount of work corresponding to the error reduction by $1/\mathrm{e}$ is the product $\mathrm{It}(\Phi)\,\mathrm{Work}(\Phi, A) \le \mathrm{It}(\Phi) C_\Phi C_A n$ (cf. (2.29)). As a characteristic quantity we choose the *effective amount of work*

$$\mathrm{Eff}(\Phi) := \mathrm{It}(\Phi) C_\Phi = -C_\Phi / \log(\rho(M)). \tag{2.31a}$$

$\mathrm{Eff}(\Phi)$ measures the amount of work for an error reduction by $1/\mathrm{e}$ in the unit '$C_A n$ arithmetic operations'. Correspondingly, the effective amount of work for the error reduction by the factor of $1/\mathrm{e}$ is given by

$$\mathrm{Eff}(\Phi, \varepsilon) := -\mathrm{It}(\Phi) C_\Phi \log(\varepsilon) = C_\Phi \log(\varepsilon)/\log(\rho(M)). \tag{2.31b}$$

Example 2.28. In the case of the model problem, the cost factor of the Gauss–Seidel iteration is $C_\Phi = 1$ (because of $C_A = 5$, cf. Remark 1.14). The numerical values in Table 1.1 suggest $\rho(M) = 0.99039$ for the grid size $h = 1/32$. Thus, the effective amount of work equals $\mathrm{Eff}(\Phi) = 103.6$. Using $\rho(M) = 0.82$ for the SOR method and $C_\Phi = 7/5$, we deduce an effective amount of work of $\mathrm{Eff}(\Phi) = 7.05$ for the SOR method with $h = 1/32$.